

WHAT IS CLAIMED IS:

- 1 1. A method for graphics processing, comprising:
2 receiving node and view data for a graphic object;
3 building a binary-space-partition tree corresponding
4 to the graphic object, the binary-space-partition
5 tree having at least one shape associated with
6 each leaf;
7 sorting shapes at each leaf of the binary-space-
8 partition tree; and
9 outputting the sorted shapes.
- 1 2. The method of claim 1, wherein the shapes are sorted
2 into a substantially back-to-front order.
- 1 3. The method of claim 1, further comprising caching the
2 shape data.
- 1 4. The method of claim 1, further comprising traversing
2 the binary-space-partition tree.
- 1 5. The method of claim 1, wherein the shapes are
2 triangles.
- 1 6. The method of claim 1, wherein a configuration
2 component is used, the configuration component
3 balancing the resolution of the binary-space-partition
4 tree against the sorting shapes at each leaf.

1 7. The method of claim 3, wherein a configuration
2 component is used, the configuration component
3 balancing resource usage against accuracy in the
4 resolution of the caching.

1 8. A method for graphics processing, comprising:
2 analyzing shapes in a graphic object;
3 creating a root node and a list of additional nodes
4 for a binary-space-partition tree, each node
5 associated with at least one shape;
6 performing a partition plane selection for each
7 additional node,
8 classifying the shapes at the additional node
9 according to the partition plane selection; and
10 creating child nodes according to the shape
11 classification.

1 9. The method of claim 8, wherein each node represents a
2 set of elements located in a 3-dimensional spatial
3 region.

1 10. The method of claim 8, wherein the shapes are
2 triangles.

- 1 11. A data processing system having at least a processor
2 and accessible memory, comprising:
3 means for receiving node and view data for a graphic
4 object;
5 means for building a binary-space-partition tree
6 corresponding to the graphic object, the binary-
7 space-partition tree having at least one shape
8 associated with each leaf;
9 means for sorting shapes at each leaf of the binary-
10 space-partition tree; and
11 means for outputting the sorted shapes.
- 1 12. The data processing system of claim 11, wherein the
2 shapes are sorted into a substantially back-to-front
3 order.
- 1 13. The data processing system of claim 11, further
2 comprising means for caching the shape data.
- 1 14. The data processing system of claim 11, further
2 comprising means for traversing the binary-space-
3 partition tree.
- 1 15. The data processing system of claim 11, wherein the
2 shapes are triangles.
- 1 16. The data processing system of claim 11, wherein a
2 configuration component is used, the configuration
3 component balancing the resolution of the binary-
4 space-partition tree against the sorting shapes at
5 each leaf.

1 17. The data processing system of claim 13, wherein a
2 configuration component is used, the configuration
3 component balancing resource usage against accuracy in
4 the resolution of the caching.

1 18. A data processing system having at least a processor
2 and accessible memory, comprising:
3 means for analyzing shapes in a graphic object;
4 means for creating a root node and a list of
5 additional nodes for a binary-space-partition
6 tree, each node associated with at least one
7 shape;
8 means for performing a partition plane selection for
9 each additional node,
10 means for classifying the shapes at the additional
11 node according to the partition plane selection;
12 and
13 means for creating child nodes according to the shape
14 classification.

1 19. The data processing system of claim 18, wherein each
2 node represents a set of elements located in a 3-
3 dimensional spatial region.

1 20. The data processing system of claim 18, wherein the
2 shapes are triangles.

- 1 21. A computer program product tangibly embodied in a
2 machine-readable medium, comprising:
3 instructions for receiving node and view data for a
4 graphic object;
5 instructions for building a binary-space-partition
6 tree corresponding to the graphic object, the
7 binary-space-partition tree having at least one
8 shape associated with each leaf;
9 instructions for sorting shapes at each leaf of the
10 binary-space-partition tree; and
11 instructions for outputting the sorted shapes.
- 1 22. The computer program product of claim 21, wherein the
2 shapes are sorted into a substantially back-to-front
3 order.
- 1 23. The computer program product of claim 21, further
2 comprising instructions for caching the shape data.
- 1 24. The computer program product of claim 21, further
2 instructions for comprising traversing the binary-
3 space-partition tree.
- 1 25. The computer program product of claim 21, wherein the
2 shapes are triangles.
- 1 26. The computer program product of claim 21, wherein a
2 configuration component is used, the configuration
3 component balancing the resolution of the binary-
4 space-partition tree against the sorting shapes at
5 each leaf.

1 27. The computer program product of claim 23, wherein a
2 configuration component is used, the configuration
3 component balancing resource usage against accuracy in
4 the resolution of the caching.

1 28. A computer program product tangibly embodied in a
2 machine-readable medium, comprising:
3 instructions for analyzing shapes in a graphic object;
4 instructions for creating a root node and a list of
5 additional nodes for a binary-space-partition
6 tree, each node associated with at least one
7 shape;
8 instructions for performing a partition plane
9 selection for each additional node,
10 instructions for classifying the shapes at the
11 additional node according to the partition plane
12 selection; and
13 instructions for creating child nodes according to the
14 shape classification.

1 29. The computer program product of claim 28, wherein each
2 node represents a set of elements located in a 3-
3 dimensional spatial region.

1 30. The computer program product of claim 28, wherein the
2 shapes are triangles.